# SESSION INITIATION PROTOCOL COMPRESSION

## Background of the Invention

The present invention pertains to communication systems and more particularly to communication systems employing Session Initiation Protocol.

Current trends in telecommunications are moving toward Internet Protocol (IP) related protocols and processes to perform tasks that used to be performed by circuit based technology. One of these internet protocol related protocols is the Session Initiation Protocol (SIP) which is used to set up, tear down and modify generic communication sessions within a communication system. See IETF RFC 2543 on "Session Initiation Protocol" 1 (SIP). SIP is a control protocol for creating, modifying and terminating communication sessions with one or more participants. These communication sessions include internet multimedia conferences, internet (or any IP network) telephone calls and multimedia distribution. SIP supports communication session descriptions that allow participants to agree on a set of compatible media types. It also supports user mobility by proxying and redirecting requests to the user's current location.

Although SIP is very generic and flexible, it presents problems in that the set up messages are extremely large relative to over-the-air signaling messages used in many cellular systems. As a result it may take a significant period of time to transfer SIP set up and negotiation messages over slow links such as RF links. The slow transfer time results in a slow set up time for communication sessions.

For example, it takes about one-half second to transfer a SIP invite message over a 9600 BPS link. This results in at least a set up time of one second if

both initiating and responding equipments are using RF links. This set up time is unacceptable for many applications such as dispatch, where the total set up time must be very fast.

Several other solutions to the problem employ using standard data compression techniques. These techniques, however, result in reduction of SIP set up messages by less than one-half. These solutions are inadequate to provide a fast set up time for SIP communication systems.

Accordingly, it would be highly advantageous to have a SIP set up message arrangement which substantially reduces the size of SIP set up messages and thereby the time required to set up SIP communications.

## Brief Description of the Drawing

FIG. 1 is a block diagram of a SIP communication in accordance with the present invention.

FIG. 2 is a message flow diagram of a registration process for SIP communication in accordance with the present invention.

FIG. 3 is a message flow diagram of a SIP invite message processed in accordance with the present invention.

FIG. 4 is a bit layout of a sample SIP invite message in accordance with the present invention.

## Description of the Preferred Embodiment

This invention conceptually pre-places data elements from a SIP header into a template located on the far end of a link between a client application and a server. The pre-placement of data elements is done prior to call set up and may for example be done during

registration or other synchronization events by a near end device such as a mobile unit. Instead of transmitting a large setup message between client and server, which is thousands of bits in length, it is replaced with a template which is less than 50 bits. The template is actually transmitted between the client and the server or vice versa. At the far end the template information is converted back into a standard setup message and the setup message accordingly handled.

FIG. 1 depicts a block diagram of an arrangement for SIP compression for system messages over limited bandwidth lengths. Client SIP application 10 requires the services of a server SIP application 30 to access the internet, for example. Typically clients of application 10, which may be a software program in a preferred embodiment, initiates a call to server SIP application 30 with a SIP invite message. This invite message includes a header that is many thousands (four to five thousand) bits of information in length. In response messages are transmitted back and forth over the limited bandwidth link 50. In a preferred embodiment, the limited bandwidth link 50 may comprise an RF (radio frequency) link, using various physical and link layer protocols known in the art.

Client SIP application 10, client SIP processing function 15, client template processing function 20 and client transport collectively may comprise a mobile unit 11. Mobile unit 11 may include such devices as mobile phones, pagers, personal digital assistants or other internet capable devices. Server 31 includes server SIP application 30, server SIP processing function 35, server template processing function 40 and server transport 45.

During registration, for example, client application 10 will communicate with server application

30 and set up a template in both the client template processing function 20 and the server template processing function 40. Client SIP application 10 generates a request for internet access to server SIP application 30. This request is transferred to client SIP processing function 15. Client SIP processing function 15 determines that this request is for server access to the internet, for example. Client SIP processing function 15 then transfers the request to the client template processing function 20. Client template processing function 20 determines that this message requires compression before being transmitted over the limited bandwidth link 50.

Client template processing function 20 determines that for the message requested by the client application 10, a particular template is required to be transmitted. Client template processing function 20 then converts or processes the requested SIP invite message, for example, to be a message as shown in FIG. 4. This message is typically 40 or 50 bits in length, although it may be as few as 18 bits long. The fields of this message will be discussed infra. Client template processing function 20 then returns the compressed message to client processing function 15. Client processing function 15 forwards the compressed message to client transport 25 for transmission over link 50 to the server 31.

Next, server transport 45 receives the compressed message from client transport 25. Server transport 45 forwards the received message to server SIP processing function 35. Server SIP processing function 35 determines that the message is a compressed message and forwards the message to server template processing function 40 for de-compression. Server template processing function then determines and retrieves the appropriate template and inserts any parameters which

may have been transmitted in the compressed message. The de-compressed message or template is then sent to SIP server processing function 35 which forwards the message to SIP server application 30. Server SIP application 30 then acts upon the message to establish the required internet access, for example.

Templates are pre-stored by the server template processing function 40. Client template processing function 20 sets up a compressed message which indicates one of the many templates to the server template processing function 40. Templates may be pre-stored as an overt action, but natural events such synchronizing address books, using the WAP browser, etc, can provide convenient times to update the templates. As new extensions are introduced into SIP standards, existing templates can be modified, or additional templates added, to support this compression method. These modifications/additions need only be introduced in the mobile client or the network, since the mobile client and network synchronize their templates. Client template processing function 20 inserts parameters that may be required for the particular message being transmitted. Server template processing function extracts these parameters from the compressed message and inserts each parameter into the de-compressed message at the appropriate locations. Effectively, the message sent by the client SIP application 10 is reconstructed by the server template processing function 40 before it is delivered to the server SIP application 30. In this way, messages sent over the limited bandwidth link 50, in a preferred embodiment an RF link, may be minimized in length and transmission time. These messages may be as small as 18 bits and approximately 40 or 50 bits may be typical. This is contrasted with the typical SIP invite message, for example, which is in the range of four to five

thousand bits. This template arrangement saves much time in transmission over the limited bandwidth link 50 and considerably reduces the setup and connect time for clients to services such as internet access.

Turning now to FIG. 2, the template set up process is shown in a message flow diagram. Typically, the client SIP application 10 pre-places the fixed template information during the registration process, that is, when the client device registers with the RF system, for example. The pre-placement is done at a time prior to the actual session set up. The pre-placement may be done when a mobile terminal is powered on, for example, or when it first registers with the RF system. Typically, the fixed template information is pre-placed by the client, but templates may also be pre-placed by the server.

Prior to setting the template, the client SIP application 10 sends a SIP register message 60 to the client SIP processing function 15. Client SIP processing function transmits this message through client transport 25 and server transport 45 to server Sip processing function 35. For the sake of brevity, client transport 25 and server transport 45 are not shown in FIG. 2. They are merely passed through functions for the purpose of message transmission between client and server or vice versa.

SIP register message 64 is then transmitted from server SIP processing function 35 to server SIP application 30. Server SIP application 30 then transmits an OK message acknowledgment 66 back to server SIP processing function 35. Server SIP processing function 35 then responds to the client SIP processing function 15 with an OK message 68.

Subsequently client's SIP application 10 then sends a set template message 72 to client SIP processing function 15. Client SIP processing function

15 sends the set template massage 74 to the client SIP template processing function 20. Client SIP template processing function 20 then responds with an OK message acknowledgment 76 to the client SIP processing function 15. Client SIP processing function 15 then transmits the set template message 78 to the server SIP processing function 35. The server SIP processing function 35 stores the template for subsequent access during the session request process. Server SIP processing function 35 then sends a set template message 80 to the server SIP template processing function 40 for storage. Server SIP template processing function 40 then responds with an OK message 82 to server SIP processing function 35. Server SIP processing function 35 then responds with an OK acknowledgment message 84 to the client's SIP processing function 15. Lastly, client SIP processing function 15 responds to the client SIP application 10 with OK acknowledgment message 86. At this point the client SIP application 10 has successfully installed a template for later use by the server.

The template can be changed at each registration process as shown in FIG. 2, although the template need not be changed on each registration. The extensions to the SIP standard can be incorporated with the need to update both the client mobile unit subscriber or the network infrastructure.

Turning now to FIG. 3, the message compression procedure is shown. For purposes of illustration a SIP invite message will be used as an example. However, this process may be applied to any of the system or application messages transmitted. Client SIP application 10 prepares and transmits the SIP invite message (header) 100 to the client SIP processing function 15. Client SIP processing function 15 determines that compression of this message is

required. Client SIP processing function 15 then transmits the SIP invite message 102 to client SIP template processing function 20. Client SIP template processing function 20 then compresses the SIP invite header into a SIP invite template 104 and transmits the template to client SIP processing function 15. Client SIP processing function 15 then transmits the SIP invite template 106 through client transport 25, link 50 and server transport 45 to server SIP processing function 35. Server SIP processing function 35 determines that the message received is a template by for example, examining a control bit in the message and transmits the SIP invite template 108 to the server SIP template processing function 40. Server SIP template processing function 40 then locates the appropriate full text message, in this case a SIP invite header message and inserts any transmitted parameters at the appropriate location and provides the SIP invite header in its uncompressed form back to server SIP processing function 35. Server SIP processing function 35 then transmits or presents the de-compressed SIP invite header message 112 to the SIP server application 30 for appropriate processing.

Server SIP application 30 then responds with an OK message 114 to server SIP processing function 35. Likewise, server SIP processing function 35 responds with an OK message 116 to acknowledge receipt and processing of the template to client SIP processing function 15. Lastly, client SIP processing function 15 responds to the client SIP application 10 with an OK message 118 indicating that the request, in this case a SIP invite header, has been received by the server and processed. The above process may be used to effectively compress whole messages or to transmit single or multiple parameters of a message. This above-described process saves the time of transmitting

complete messages over the limited bandwidth link 50 which may in the preferred embodiment be an RF link.

The processing ability of the client server is substituted for the transmission time over link 50, thereby requiring a minimum of real time transmission and allowing SIP set ups to occur very rapidly instead of taking many seconds to accomplish.

FIG. 4 depicts a bit map of a sample template which is transmitted between client and server. The example shown in FIG. 4 is a SIP invite template 200. SIP invite template 200 includes a service request message identifier data field 201. This field identifies the kind or type of message, in this case a SIP invite message. In a preferred embodiment of the invention this field will be approximately eight bits in length.

The next field of SIP invite message 200 is the template on/off flag data field 202. This field is a one bit field which indicates whether the particular message is or is not a template. If this field is set to one, the message is a template and if the field is reset to zero, the message is not a template.

The next data field of the SIP invite message 200 is the template index 203. The template index is a number which tells the far end, typically the server, which particular pre-placed template is to be reconstructed. In a preferred embodiment, this data field is eight bits in length although larger amounts of bits may be required to distinguish templates.

The last data field of SIP invite message 200 is the additional parameters present flag 204. This data field is a one bit field which indicates when set to one that additional parameters are included within optional field 205 which are the additional parameters. When field 204 is set to zero, it indicates that no additional parameters are present.

As can be seen, the message of FIG. 4 actually transmitted between client and server or vice versa via the link 50 is quite short and may be as short as 18 bits in length. Typical SIP invite messages are, for example, four or five thousand bits in length and require several seconds for transmission and acknowledgements to proceed. In a single template message the information may be transmitted and acknowledged in hundredths or thousandths of a second instead of several seconds. Thereby the setup time for client access to such services as session based communication through the internet may be virtually transparent and does not impede links such as wireless links connecting clients and servers.

Although the preferred embodiment of the invention has been illustrated, and that form described in detail, it will be readily apparent to those skilled in the art that various modifications may be made therein without departing from the spirit of the present invention or from the scope of the appended claims.